MAINTAINING LIMITED-RANGE CONNECTIVITY
AMONG SECOND-ORDER AGENTS*

KETAN SAVLA†, GIUSEPPE NOTARSTEFANO‡ AND FRANCESCO BULLO†

Abstract. In this paper we consider ad-hoc networks of robotic agents with double integrator dynamics. For such networks, the connectivity maintenance problems are: (i) do there exist control inputs for each agent to maintain network connectivity, and (ii) given desired controls for each agent, can one compute the closest connectivity-maintaining controls in a distributed fashion? The proposed solution is based on three contributions. First, we define and characterize admissible sets for double integrators to remain inside disks. Second, we establish an existence theorem for the connectivity maintenance problem by introducing a novel state-dependent graph, called the *double-integrator disk graph*. Finally, we design a distributed "flow-control" algorithm to compute optimal connectivity-maintaining controls.

1. Introduction. This work is a contribution to the emerging discipline of motion coordination for ad-hoc networks of mobile autonomous agents. This loose terminology refers to groups of robotic agents with limited mobility and communication capabilities. In the not too distant future, these groups of coordinated devices will perform a variety of challenging tasks including search and recovery operations, surveillance, exploration and environmental monitoring. The potential advantages of employing arrays of agents have recently motivated vast interest in this topic. For example, from a control viewpoint, a group of agents inherently provides robustness to failures of single agents or of communication links.

The motion coordination problem for groups of autonomous agents is a control problem in the presence of communication constraints. Typically, each agent makes decisions based only on partial information about the state of the entire network that is obtained via communication with its immediate neighbors. One important difficulty is that the topology of the communication network depends on the agents' locations and, therefore, changes with the evolution of the network. In order to ensure a desired emergent behavior for a group of agents, it is necessary that the group does not disintegrate into subgroups that are unable to communicate with each other. In other words, some restrictions must be applied on the movement of the agents to ensure *connectivity* among the members of the group. In terms of design, it is required to constrain the control input such that the resulting topology maintains connectivity throughout its course of evolution. In [2], a connectivity constraint was developed for a group of agents modeled as first-order discrete time dynamic systems. In [2] and in the related references [3, 4], this constraint is used to solve rendezvous problems. Connectivity constraints for line-of-sight communication are proposed in [5]. Another approach to connectivity maintenance for first-order systems is proposed in [6]. In [7], a centralized procedure to find the set of control inputs that maintain

[2]Center for Control, Dynamical Systems and Computation, University of California, Santa Barbara, CA 93106, United States, `ketansavla@umail.ucsb.edu, bullo@engineering.ucsb.edu`, `http://motion.mee.ucsb.edu`
[3]Department of Information Engineering, Università di Padova, Via Gradenigo 6/a, 35131 Padova, Italy, `notarste@dei.unipd.it`

k-hop connectivity for a network of agents is given. However, there is no guarantee that the resulting set of feasible control inputs in non-empty. In this paper we fully characterize the set of admissible control inputs for a group of agents modeled as second order discrete time dynamic systems, which ensures connectivity of the group in the same spirit as described earlier.

The contributions of the paper are threefold. First, we consider a control system consisting of a double integrator with bounded control inputs. For such a system, we define and characterize the admissible set that allows the double integrator to remain inside disks. Second, we define a novel state-dependent graph – the *double-integrator disk graph* – and give an existence theorem for the connectivity maintenance problem for networks of second order agents with respect to an appropriate version of this new graph. Finally, we consider a relevant optimization problem, where given a set of desired control inputs for all the agents it is required to find the optimal set of connectivity-maintaining control inputs. We cast this problem into a standard quadratic programming problem and provide a distributed "flow-control" algorithm to solve it.

The paper is organized as follows. In Section 2, we define and characterize the admissible sets for a double integrator to remain inside a disk and based on this we define a new graph – the *double-integrator disk graph*. In Section 3, we provide an existence theorem for the set of control inputs for the whole network of second order agents that maintains connectivity with respect to an appropriately scaled version of this new graph. We also characterize and give an inner polytopic representation of the constraint set for these connectivity-maintaining control inputs. In Section 4, we consider the problem of searching this constraint set for the optimal set of controls in a distributed way. Section 5 has some illustrative simulations which also suggest an alternative way of achieving a weak form of flocking of the agents. Finally we conclude with a few remarks about future work in Section 6.

2. Preliminary developments. We begin with some notations. We let $\mathbb{N}$, $\mathbb{N}_0$, and $\mathbb{R}_+$ denote the natural numbers, the non-negative integer numbers, and the positive real numbers, respectively. For $d \in \mathbb{N}$, we let 0_d and 1_d denote the vectors in $\mathbb{R}^d$ whose entries are all 0 and 1, respectively. We let $\|p\|$ denote the Euclidean norm of $p \in \mathbb{R}^d$. For $r \in \mathbb{R}_+$ and $p \in \mathbb{R}^d$, we let $B(p, r)$ denote the closed ball centered at p with radius r, i.e., $B(p, r) = \{q \in \mathbb{R}^d \mid \|p - q\| \leq r\}$. For $x, y \in \mathbb{R}^d$, we let $x \preceq y$ denote component-wise inequality, i.e., $x_k \leq y_k$ for $k \in \{1, \ldots, d\}$. We let $f : A \rightrightarrows B$ denote a set-valued map; in other words, for each $a \in A$, $f(a)$ is a subset of B. We identify $\mathbb{R}^d \times \mathbb{R}^d$ with $\mathbb{R}^{2d}$.

2.1. Maintaining a double integrator inside a disk. For $t \in \mathbb{N}_0$, consider the discrete-time control system in $\mathbb{R}^{2d}$

$$
\begin{aligned}
p[t + 1] &= p[t] + v[t], \\
v[t + 1] &= v[t] + u[t],
\end{aligned}
\tag{2.1}
$$

where the norm of the control is upper-bounded by $r_{\mathrm{ctr}} \in \mathbb{R}_+$, i.e., $u[t] \in B(0_d, r_{\mathrm{ctr}})$ for $t \in \mathbb{N}_0$. We refer to this control system as the *discrete-time double integrator* in $\mathbb{R}^d$ or, more loosely, as a second-order system. Given $(p, v) \in \mathbb{R}^{2d}$ and $\{u_\tau\}_{\tau \in \mathbb{N}_0} \subseteq B(0_d, r_{\mathrm{ctr}})$, let $\phi(t, (p, v), \{u_\tau\})$ denote the solution of (2.1) at time $t \in \mathbb{N}_0$ from initial condition (p, v) with inputs $u_1, \ldots, u_{t-1}$.

In what follows we consider the following problem: assume that the initial position of (2.1) is inside a disk centered at 0_d, find inputs that keep it inside that disk. This

2

task is impossible for general values of the initial velocity. In what follows we identify assumptions on the initial velocity that render the task possible.

For $r_{\mathrm{pos}} \in \mathbb{R}_+$, we define the *admissible set at time zero* by

$$\mathcal{A}_0^d(r_{\mathrm{pos}}) = B(0_d, r_{\mathrm{pos}}) \times \mathbb{R}^d.$$

For $r_{\mathrm{pos}}, r_{\mathrm{ctr}} \in \mathbb{R}_+$, we define the *admissible set for m time steps* by

$$\mathcal{A}_m^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) = \big\{ (p, v) \in \mathbb{R}^{2d} \mid \exists \{u_\tau\}_{\tau \in [0, m-1]} \subseteq B(0_d, r_{\mathrm{ctr}})$$
$$\text{s.t. } \phi(t, (p, v), \{u_\tau\}) \in \mathcal{A}_0^d(r_{\mathrm{pos}}) \ \forall t \in [0, m] \big\},$$

and the *admissible set* by

$$\mathcal{A}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) = \big\{ (p, v) \in \mathbb{R}^{2d} \mid \exists \{u_\tau\}_{\tau \in \mathbb{N}_0} \subseteq B(0_d, r_{\mathrm{ctr}})$$
$$\text{s.t. } \phi(t, (p, v), \{u_\tau\}) \in \mathcal{A}_0^d(r_{\mathrm{pos}}), \ \forall t \in \mathbb{N}_0 \big\}.$$

With slight abuse of notation we shall sometimes suppress the arguments in the definitions of admissible sets. The following theorem establishes some important properties of the admissible sets.

THEOREM 2.1 (Properties of the admissible sets). *For all $d \in \mathbb{N}$ and $r_{\mathrm{pos}}, r_{\mathrm{ctr}} \in \mathbb{R}_+$, the following statements hold:*

(i) *for all $m \in \mathbb{N}$, $\mathcal{A}_m^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) \subseteq \mathcal{A}_{m-1}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ and*

$$\mathcal{A}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) = \lim_{m \to +\infty} \mathcal{A}_m^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) = \lim_{m \to +\infty} \cap_{k=1}^m \mathcal{A}_k^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) ;$$

(ii) $\mathcal{A}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ *is a convex, compact set and is the largest controlled-invariant[1] subset of $\mathcal{A}_0^d(r_{\mathrm{pos}})$;*

(iii) $\mathcal{A}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ *is invariant under orthogonal transformations in the sense that, if $(p, v) \in \mathcal{A}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$, then also $(Rp, Rv) \in \mathcal{A}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ for all orthogonal[2] matrices R in $\mathbb{R}^{d \times d}$;*

(iv) *if $0 < r_1 < r_2$, then $\mathcal{A}^d(r_{\mathrm{pos}}, r_1) \subset \mathcal{A}^d(r_{\mathrm{pos}}, r_2)$ and $\mathcal{A}^d(r_1, r_{\mathrm{ctr}}) \subset \mathcal{A}^d(r_2, r_{\mathrm{ctr}})$.*

Proof. The two facts in statement (i) are direct consequences of the definitions of $\mathcal{A}_m^d$ and $\mathcal{A}^d$. Regarding statement (ii), each $\mathcal{A}_m^d$, $m \in \mathbb{N}$, is closed, the intersection of closed sets is closed, and, therefore, $\mathcal{A}^d = \lim_{m \to +\infty} \cap_{k=1}^m \mathcal{A}_k^d$ is closed. To show that $\mathcal{A}^d$ is bounded it suffices to show that $\mathcal{A}_1^d$ is bounded. Note that $(p, v) \in \mathcal{A}_1^d$ implies that there exists $u \in B(0_d, r_{\mathrm{ctr}})$ such that $(p, v) \in \mathcal{A}_0^d$ and $(p + v, v + u) \in \mathcal{A}_0^d$. This, in turn, implies that $p \in B(0_d, r_{\mathrm{pos}})$ and $p + v \in B(0_d, r_{\mathrm{pos}})$. Therefore, $\mathcal{A}_1^d$ is bounded. Next, we prove that $\mathcal{A}_m^d$ is convex. Given (p_1, v_1) and (p_2, v_2) in $\mathcal{A}_m^d$, let u_1 and u_2 be controls that ensure that $\phi(t, (p_i, v_i), \{u_i\}) \in \mathcal{A}_0^d$, $t \in [0, m]$, $i \in \{1, 2\}$. For $\lambda \in [0, 1]$, consider the initial condition $(p_\lambda, v_\lambda) = (\lambda p_1 + (1 - \lambda)p_2, \lambda v_1 + (1 - \lambda)v_2)$ and the input $u_\lambda = \lambda u_1 + (1 - \lambda)u_2$, and note that, by linearity,

$$\phi(t, (p_\lambda, v_\lambda), u_\lambda) = \lambda \phi(t, (p_1, v_1), \{u_1\}) + (1 - \lambda)\phi(t, (p_2, v_2), \{u_2\}), \quad t \in [0, m].$$

Because $\phi(t, (p_1, v_1), \{u_1\})$ and $\phi(t, (p_2, v_2), \{u_2\})$ belong to the convex set $\mathcal{A}_0^d$, then also their convex combination does. Therefore, (p_λ, v_λ) belongs to $\mathcal{A}_m^d$, and $\mathcal{A}_m^d$ is convex. Finally, $\mathcal{A}^d$ is convex because the intersection of convex sets is convex.

---

[1] A set is controlled invariant for a control system if there exists a feedback law such that the set is positively invariant for the closed-loop system.

[2] A matrix $R \in \mathbb{R}^{d \times d}$ is orthogonal if $RR^T = R^T R = I_d$.

Next, we show that $\mathcal{A}^d$ is controlled invariant. Given $(p, v) \in \mathcal{A}^d$ (with corresponding control sequence $\{u_\tau\}_{\tau \in \mathbb{N}_0}$), we need to show that there exists a control input $x \in B(0_d, r_{\text{ctr}})$ such that $\phi(1, (p, v), x) \in \mathcal{A}^d$. Such input can be chosen as $x = u_0$. Indeed, the control sequence $\{u_{\tau+1}\}_{\tau \in \mathbb{N}_0}$ keeps the trajectory starting from $\phi(1, (p, v), x)$ inside $\mathcal{A}_0^d$ and, therefore, $\phi(1, (p, v), x) \in \mathcal{A}^d$. Additionally, it is easy to see that $\mathcal{A}^d \subset \mathcal{A}_0^d$. Finally, we need to prove that $\mathcal{A}^d$ is the largest controlled-invariant subset of $\mathcal{A}_0^d$. Assume that there exists $\mathcal{A}^{d*}$ with the same properties and larger than $\mathcal{A}^d$. This means that there exists $(p, v) \in \mathcal{A}^{d*} \setminus \mathcal{A}^d$. This is equivalent to saying that $\exists\, \tau^* \in \mathbb{N}_0$ such that, for every choice of the input u, $\phi(\tau^*, (p, v), u) \notin \mathcal{A}_0^d$. But, since $\mathcal{A}^{d*} \subset \mathcal{A}_0^d$, this leads to a contradiction.

Regarding statement (iii), observe that, if $(p, v) \in \mathcal{A}_0^d$, then $(Rp, Rv) \in \mathcal{A}_0^d$ and, if $u \in B(0, r_{\text{ctr}})$, then $Ru \in B(0, r_{\text{ctr}})$. Therefore, using again the linearity of the maps ϕ, the proof follows. Regarding statement (iv), the two results follow from the definition of $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ and the facts that, for all $0 < r_1 < r_2$, $B(0, r_1) \subset B(0, r_2)$ and $\mathcal{A}_0^d(r_1) \subset \mathcal{A}_0^d(r_2)$. $\square$

Next, we study the set-valued map that associates to each state in $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ the set of control inputs that keep the state inside $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ in one step. We define the *admissible control set* $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) : \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}}) \rightrightarrows B(0_d, r_{\text{ctr}})$ by

$$\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v) = \{u \in B(0_d, r_{\text{ctr}}) \mid (p + v, v + u) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})\},$$

or, equivalently,

$$\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v) = B(0_d, r_{\text{ctr}}) \cap \{w - v \mid (p + v, w) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})\}. \quad (2.2)$$

LEMMA 2.2 (Properties of the admissible control set). *For all $(p, v) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, the set $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$ is non-empty, convex and compact. For generic $(p, v) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, the set $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$ does not contain 0_d.*

Proof. The non-emptiness of the set $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$ derives directly from the definition of $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$. Clearly, from equation (2.2), $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$ is closed (it is the intersection of two closed sets). It is also bounded ($\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v) \subset B(0_d, r_{\text{ctr}})$), hence it is compact. To prove that it is convex, we need to show the following: given $(p, v) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, if there exist u_1 and u_2 in $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$ such that $\phi(1, (p, v), u_1)$ and $\phi(1, (p, v), u_2)$ belong to $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, then $u_{12} = \lambda u_1 + (1 - \lambda) u_2$, $\lambda \in [0, 1]$, belongs to $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$, that is, $\phi(1, (p, v), u_{12}) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$. But this fact follows directly from the linearity of ϕ and the convexity of $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$. This proves that $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$ is convex. The fact that it does not necessarily contain the origin can be proven by contradiction as follows. Consider a $(p, v) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ such that $v \neq 0_d$ and $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$ contains 0_d. This means that $(p + v, v)$ also belongs to $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$. Now, either $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p + v, v)$ does not contain 0_d, in which case we have proved the statement, or $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ also contains $(p + 2v, v)$. Continuing along these lines, if it were true that $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p, v)$ contains the origin for all $(p, v) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, then one could show that $(p + tv, v)$ belongs to $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ for all $t \in \mathbb{N}$. However, $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ is bounded by Theorem 2.1. Hence, one can always find a $t^* \in \mathbb{N}$ such that $(p + t^*v, v) \in \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$ but $(p + (t^* + 1)v, v) \notin \mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$, thereby proving that $\mathcal{U}^d(r_{\text{pos}}, r_{\text{ctr}}) \cdot (p + t^*v, v)$ does not contain 0_d. $\square$

2.2. Computing admissible sets. We characterize $\mathcal{A}^d$ for $d = 1$ in the following result and we illustrate the outcome in Figure 2.1.

LEMMA 2.3 (Admissible set in 1 dimension). *For $r_{\text{pos}}, r_{\text{ctr}} \in \mathbb{R}_+$, the following holds:*

(i) $\mathcal{A}^1(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ is the polytope containing the points $(p, v) \in \mathbb{R}^2$ satisfying

$$-\frac{r_{\mathrm{pos}}}{m} - \frac{m-1}{2}r_{\mathrm{ctr}} \leq v + \frac{p}{m} \leq \frac{r_{\mathrm{pos}}}{m} + \frac{m-1}{2}r_{\mathrm{ctr}}, \qquad (2.3)$$

for all $m \in \mathbb{N}$, and $p \in [-r_{\mathrm{pos}}, r_{\mathrm{pos}}]$;

(ii) If $\widehat{m}(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) \in \mathbb{N}$ is defined by

$$\widehat{m}(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) = \left\lceil -\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{4r_{\mathrm{pos}}}{r_{\mathrm{ctr}}}} \right\rceil, \qquad (2.4)$$

then $\mathcal{A}^1 = \mathcal{A}_m^1 = \mathcal{A}_{\widehat{m}(r_{\mathrm{pos}}, r_{\mathrm{ctr}})}^1$, for $m \geq \widehat{m}(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$.

Proof. Regarding statement (i), it suffices to show that, for $m \in \mathbb{N}$, $\mathcal{A}_m^1(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ is the set of points in $\mathcal{A}_{m-1}^1(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ that satisfy equation (2.3). If we show that this property holds for all m, then we can use statement (i) of Theorem 2.1 to complete the proof. Consider the set of equations (2.1) for m consecutive time indices after t. The solution of the linear system can be written in terms of the state at instant t as

$$\begin{bmatrix} p[t+m] \\ v[t+m] \end{bmatrix} = \begin{bmatrix} 1 & m \\ 0 & 1 \end{bmatrix}\begin{bmatrix} p[t] \\ v[t] \end{bmatrix} + \sum_{\tau=0}^{m-1} \begin{bmatrix} 1 & (m-1-\tau) \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} u[t+\tau], \qquad (2.5)$$

where we used the equality

$$A^\tau = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}^\tau = \begin{bmatrix} 1 & \tau \\ 0 & 1 \end{bmatrix}, \qquad \tau \in \mathbb{N}.$$

It is clear that the points on the boundary of $\mathcal{A}_m^1$ have the property that the maximum control effort is needed to enforce the constraint. In other words we look for the points $(p[t], v[t]) \in \mathcal{A}_0^1$ with $v[t] \geq 0$ (the case $v[t] \leq 0$ can be solved in a similar way) such that the points $p[t+m] \leq r_{\mathrm{cmm}}$ are reached by using the maximum control effort $u[t+\tau] = -r_{\mathrm{ctr}}$, $\tau \in \{0, \ldots, m-1\}$.

Substituting the expression of the control in (2.5) we obtain

$$p[t+m] = p[t] + mv[t] - r_{\mathrm{ctr}}\sum_{\tau=0}^{m-1}(m-1-\tau),$$

$$v[t+m] = v[t] - mr_{\mathrm{ctr}},$$

and using the equality $\sum_{\tau=0}^{m-1}(m-1-\tau) = \frac{m(m-1)}{2}$, we have

$$\begin{aligned} p[t+m] &= p[t] + mv[t] - r_{\mathrm{ctr}}\frac{m(m-1)}{2}, \\ v[t+m] &= v[t] - mr_{\mathrm{ctr}}, \end{aligned} \qquad (2.6)$$

In order to belong to $\mathcal{A}_m^1$, the point $(p[t], v[t])$ must satisfy the constraint $p[t+\tau] \leq r_{\mathrm{cmm}}$, $\tau \in \{1, \ldots, m\}$, or equivalently

$$v[t] \leq -\frac{p[t]}{\tau} + \frac{r_{\mathrm{cmm}}}{\tau} + r_{\mathrm{ctr}}\frac{(\tau-1)}{2}, \qquad \tau \in \{1, \ldots, m\}.$$

Using the same procedure for the points in the half plane $v[t] \leq 0$ (in this case the control is $u[t+\tau] = r_{\mathrm{ctr}}$, $\tau \in \{0, \ldots, m-1\}$), it turns out that $\mathcal{A}_m^1$ is equal to the set of all pairs $(p, v) \in \mathcal{A}_0^1$ satisfying

$$-\frac{p}{\tau} - \frac{r_{\mathrm{cmm}}}{\tau} - \frac{\tau-1}{2}r_{\mathrm{ctr}} \leq v \leq -\frac{p}{\tau} + \frac{r_{\mathrm{cmm}}}{\tau} + \frac{\tau-1}{2}r_{\mathrm{ctr}}, \qquad \tau \in \{1, \ldots, m\}.$$

5

By using statement (i) of Theorem 2.1 the proof is complete.

Regarding statement (ii), let us consider the case $v[t] \geq 0$ and evaluate the points on the boundary such that $(p[t+m], v[t+m]) = (r_{\mathrm{cmm}}, 0)$, $m \in \mathbb{N}$. These points are obtained by substituting the above value of $(p[t+m], v[t+m])$ in (2.6). The points obtained are (p, v) such that

$$p = r_{\mathrm{cmm}} - m\frac{(m+1)}{2}r_{\mathrm{ctr}}, \qquad m \in \mathbb{N}_0.$$

It is easy to see that $\widehat{m}(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$, as defined in equation (2.4), is the lowest m such that $p \leq -r_{\mathrm{cmm}}$. $\square$
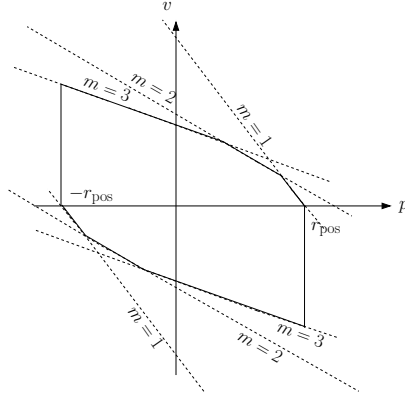

Fig. 2.1: The admissible set $\mathcal{A}^1$ for generic values of r_{pos} and r_{ctr}

REMARKS 2.4.
(i) If $r_{\mathrm{ctr}} \geq 2r_{\mathrm{pos}}$, then $\mathcal{A}^1 = \mathcal{A}_1^1$, that is, for sufficiently large $r_{\mathrm{ctr}}/r_{\mathrm{pos}}$, the admissible set is equal to the admissible set in 1 time step.
(ii) The methodology for constructing $\mathcal{A}^1(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ is related to the procedure for constructing the so-called isochronic regions for discrete time optimal control of double integrators, as outlined in [8]. $\square$

Next, we introduce some definitions useful to provide an inner approximation of $\mathcal{A}^d$ when $d \geq 2$. Given $p \in \mathbb{R}^d$ and $v \in \mathbb{R}^d \setminus \{0_d\}$, define $p_\parallel \in \mathbb{R}$ and $p_\perp \in \mathbb{R}^d$ by

$$p = p_\parallel \frac{v}{\|v\|} + p_\perp,$$

where $p_\perp \cdot v = 0$. For $r_{\mathrm{pos}}, r_{\mathrm{ctr}} \in \mathbb{R}_+$, define

$$\mathcal{A}_\parallel^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}}) = \Big\{ (p, v) \in B(0_d, r_{\mathrm{pos}}) \times \mathbb{R}^d \mid v = 0_d \text{ or}$$

$$(p_\parallel, \|v\|) \in \mathcal{A}^1\big(\sqrt{r_{\mathrm{pos}}^2 - \|p_\perp\|^2}, r_{\mathrm{ctr}}\big)\Big\}. \quad (2.7)$$

LEMMA 2.5. For $r_{\mathrm{pos}}, r_{\mathrm{ctr}} \in \mathbb{R}_+$, $\mathcal{A}_\parallel^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$ is a compact subset of $\mathcal{A}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}})$.

Proof. We begin by showing that definition (2.7) is equivalent to

$$\mathcal{A}_{\|}^d(r_{\text{pos}}, r_{\text{ctr}}) = \Big\{ (p, v) \in \mathcal{A}_0^d \mid v = 0_d \text{ or } \exists \{u_{\|\tau}\}_{\tau \in \mathbb{N}_0} \subseteq [-r_{\text{ctr}}, r_{\text{ctr}}]$$

$$\text{s.t. } \phi\Big(t, (p, v), \{u_{\|\tau}\}\frac{v}{\|v\|}\Big) \in \mathcal{A}_0^d(r_{\text{pos}}), \ \forall t \in \mathbb{N}_0 \Big\}. \quad (2.8)$$

To establish this equivalence, we use the definition of the set $\mathcal{A}^1$. For $v \neq 0_d$, we rewrite the solution of the system as

$$\phi(t, (p, v), \{u_\tau\}) = \phi_{\|}(t, (p, v), \{u_\tau\})\frac{v}{\|v\|} + \phi_{\perp}(t, (p, v), \{u_\tau\}),$$

where $\phi_{\perp}(t, (p, v), \{u_\tau\}) \cdot v = 0$ for all $t \in \mathbb{N}_0$. It is easy to see that, if $\{u_\tau\}_{\tau \in \mathbb{N}_0} = \{u_{\|\tau}\}_{\tau \in \mathbb{N}_0}\frac{v}{\|v\|}$, then $\phi_{\perp}(t, (p, v), \{u_\tau\}) = (p_{\perp}, 0_d)$ for all $t \in \mathbb{N}_0$. Therefore,

$$\phi(t, (p, v), \{u_\tau\}) = \phi_{\|}(t, (p, v), \{u_\tau\})\frac{v}{\|v\|} + (p_{\perp}, 0_d).$$

Note that, if $p = p_{\|}\frac{v}{\|v\|} + p_{\perp}$, then $\|p\| \leq r_{\text{pos}}$ if and only if $p_{\|} \leq \sqrt{r_{\text{pos}}^2 - \|p_{\perp}\|^2}$. Therefore, the property $\phi\Big(t, (p, v), \{u_{\|\tau}\}\frac{v}{\|v\|}\Big) \in \mathcal{A}_0^d(r_{\text{pos}})$ is equivalent to

$$\phi_{\|}\Big(t, (p, v), \{u_{\|\tau}\}\frac{v}{\|v\|}\Big) \in \mathcal{A}_0^1\Big(\sqrt{r_{\text{pos}}^2 - \|p_{\perp}\|^2}\Big),$$

and, in turn, definitions (2.7) and (2.8) are equivalent. In order to prove that $\mathcal{A}_{\|}^d(r_{\text{pos}}, r_{\text{ctr}})$ is compact, we simply observe that it is a closed subset of the compact set $\mathcal{A}^d(r_{\text{pos}}, r_{\text{ctr}})$. ☐

REMARK 2.6. *In what follows we use our representation of $\mathcal{A}_{\|}^d$ to compute an inner approximation for the convex set $\mathcal{A}^d$, for $d \geq 2$. For example, for fixed $p \in B(0_d, r_{\text{pos}})$, we compute velocity vectors v such that $(p, v) \in \mathcal{A}^d$ by considering a sample of unit-length vectors $w \in \mathbb{R}^d$ and computing the maximum allowable velocity v parallel to w according to equation (2.7). Furthermore, we perform computations by adopting inner polytopic representations for the various compact convex sets.* ☐

2.3. The double-integrator disk graph. Let us introduce some concepts about state dependent graphs and some useful examples. For a set X, let $\mathbb{F}(X)$ be the collection of finite subsets of X; e.g., $\mathcal{P} \in \mathbb{F}(\mathbb{R}^d)$ is a set of points. For a finite set X, let $\mathbb{G}(X)$ be the set of undirected graphs whose vertices are elements of X, i.e., whose vertex set belongs to $\mathbb{F}(X)$. For a set X, a *state dependent graph on X* is a map $\mathcal{G} : \mathbb{F}(X) \to \mathbb{G}(X)$ that associates to a finite subset V of X an undirected graph with vertex set V and edge set $\mathcal{E}_{\mathcal{G}}(V)$ where $\mathcal{E}_{\mathcal{G}} : \mathbb{F}(X) \to \mathbb{F}(X \times X)$ satisfies $\mathcal{E}_{\mathcal{G}}(V) \subseteq V \times V$. In other words, what edges exist in $\mathcal{G}(V)$ depends on the elements of V that constitute the nodes.

The following three examples of state dependent graphs play an important role. First, given $r_{\text{pos}} \in \mathbb{R}_+$, the *disk graph* $\mathcal{G}_{\text{disk}}(r_{\text{pos}})$ is the state dependent graph on $\mathbb{R}^d$ defined as follows: for $\{p_1, \ldots, p_n\} \subset \mathbb{R}^d$, the pair (p_i, p_j) is an edge in $\mathcal{G}_{\text{disk}}(r_{\text{pos}}) \cdot (\{p_1, \ldots, p_n\})$ if and only if

$$\|p_i - p_j\| \leq r_{\text{pos}} \quad \Longleftrightarrow \quad p_i - p_j \in B(0_d, r_{\text{pos}}).$$

Second, given $r_{\text{pos}}, r_{\text{ctr}} \in \mathbb{R}_+$, the *double-integrator disk graph* $\mathcal{G}_{\text{di-disk}}(r_{\text{pos}}, r_{\text{ctr}})$ is the state dependent graph on $\mathbb{R}^{2d}$ defined as follows: for $\{(p_1, v_1), \ldots, (p_n, v_n)\} \subset \mathbb{R}^{2d}$,

7

the pair $((p_i, v_i), (p_j, v_j))$ is an edge if and only if the relative positions and velocities satisfy

$$(p_i - p_j, v_i - v_j) \in \mathcal{A}^d(r_{\mathrm{pos}}, r_{\mathrm{ctr}}).$$

Third, it is convenient to define the disk graph also as a state dependent graph on $\mathbb{R}^{2d}$ by stating that $((p_i, v_i), (p_j, v_j))$ is an edge if and only if (p_i, p_j) is an edge of the disk graph on $\mathbb{R}^d$. We illustrate the first two graphs in Figure 2.2.
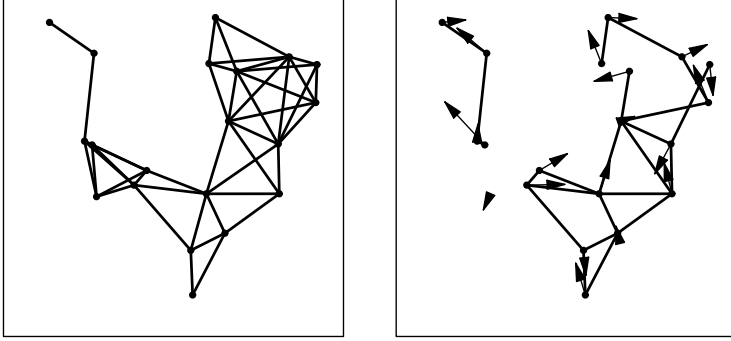

Fig. 2.2: The disk graph and the double-integrator disk graph in $\mathbb{R}^2$ for 20 agents with random positions and velocities.

REMARK 2.7. *As is well known, the disk graph is invariant under rigid transformations and reflections. Loosely speaking, the double integrator disk graph is invariant under the following class of transformations: position and velocities of the agents may be expressed with respect to any rotated and translated frame that is moving at constant linear velocity. These transformations are related to the classic Galilean transformations in geometric mechanics.* □

3. Connectivity constraints among second-order agents. In this section we state the model, the notion of connectivity, and a sufficient condition that guarantees connectivity can be preserved.

3.1. Networks of robotic agents with second-order dynamics and the connectivity maintenance problem. We begin by introducing the notion of *network of robotic agents with second-order dynamics* in $\mathbb{R}^d$. Let n be the number of agents. Each agent has the following computation, motion control, and communication capabilities. For $i \in \{1, \ldots, n\}$, the ith agent has a processor with the ability of allocating continuous and discrete states and performing operations on them. The ith agent occupies a location $p_i \in \mathbb{R}^d$, moves with velocity $v_i \in \mathbb{R}^d$, according to the discrete-time double integrator dynamics in (2.1), i.e.,

$$\begin{aligned} p_i[t+1] &= p_i[t] + v_i[t], \\ v_i[t+1] &= v_i[t] + u_i[t], \end{aligned} \tag{3.1}$$

where the norm of all controls $u_i[t]$, $i \in \{1, \ldots, n\}$, $t \in \mathbb{N}_0$, is upper-bounded by $r_{\mathrm{ctr}} \in \mathbb{R}_+$. The communication model is the following. The processor of each agent has access to the agent location and velocity. Each agent can transmit information

to other agents within a distance $r_{\mathrm{cmm}} \in \mathbb{R}_+$. We remark that the control bound r_{ctr} and the communication radius r_{cmm} are the same for all agents.

REMARKS 3.1.
 (i) *Our network model assumes synchronous execution, although it would be important to consider more general asynchronous networks.*
 (ii) *We will not address in this paper the correctness of our algorithms in the presence of measurement errors or communication quantization.* $\quad\square$

We now state the control design problem of interest.

PROBLEM 3.2 (Connectivity Maintenance). *Choose a state dependent graph $\mathcal{G}_{\mathrm{target}}$ on $\mathbb{R}^{2d}$ and design (state dependent) control constraints sets with the following property: if each agent's control takes values in the control constraint set, then the agents move in such a way that the number of connected components of $\mathcal{G}_{\mathrm{target}}$ (evaluated at the agents' states) does not increase with time.* $\quad\square$

This objective is to be achieved with the limited information available through message exchanges between agents. This problem was originally stated and solved for first-order agents in [2].

3.2. A known result for agents with first-order dynamics. In [2], a connectivity constraint was developed for a set of agents modeled by first-order discrete-time dynamics:

$$p_i[t+1] = p_i[t] + u_i[t].$$

Here the graph whose connectivity is of interest, is the disk graph $\mathcal{G}_{\mathrm{disk}}(r_{\mathrm{cmm}})$ over the vertices $\{p_1[t], \ldots, p_n[t]\}$. Network connectivity is maintained by restricting the allowable motion of each agent. In particular, it suffices to restrict the motion of each agent as follows. If agents i and j are neighbors in the r_{cmm}-disk graph $\mathcal{G}_{\mathrm{disk}}(r_{\mathrm{cmm}})$ at time t, then their positions at time $t+1$ are required to belong to $B\left(\frac{p_i[t]+p_j[t]}{2}, \frac{r_{\mathrm{cmm}}}{2}\right)$. In other words, connectivity between i and j is maintained if

$$u_i[t] \in B\left(\frac{p_j[t] - p_i[t]}{2}, \frac{r_{\mathrm{cmm}}}{2}\right),$$
$$u_j[t] \in B\left(\frac{p_i[t] - p_j[t]}{2}, \frac{r_{\mathrm{cmm}}}{2}\right).$$
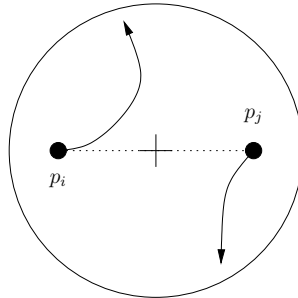
The constraint is illustrated in Figure 3.1.

Fig. 3.1: Starting from p_i and p_j, the agents are restricted to move inside the disk centered at $\frac{p_i+p_j}{2}$ with radius $\frac{r_{\mathrm{cmm}}}{2}$.

Note that this constraint takes into account only the positions of the agents; this fact can be attributed to the *first-order dynamics* of the agents. We wish to construct a similar constraint for agents with second order dynamics. It is reasonable to expect that this new constraint will depend on positions as well as velocities of the neighboring agents.

3.3. An appropriate graph for the connectivity maintenance problem for agents with second-order dynamics.
We begin working on the stated problem with a negative result regarding two candidate target graphs.

LEMMA 3.3 (Unsuitable graphs). *Consider a network of n agents with double integrator dynamics (3.1) in $\mathbb{R}^d$. Let r_{cmm} be the communication range and let r_{ctr} be the control bound. Let $\mathcal{G}_{\mathrm{target}}$ be either $\mathcal{G}_{\mathrm{disk}}(r_{\mathrm{cmm}})$ on $\mathbb{R}^{2d}$ or $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, 2r_{\mathrm{ctr}})$. There exist states $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$ such that*

(i) the graph $\mathcal{G}_{\mathrm{target}}$ at $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$ is connected, and

(ii) for all $\{u_i\}_{i \in \{1,\ldots,n\}} \subseteq B(0_d, r_{\mathrm{ctr}})$, the graph $\mathcal{G}_{\mathrm{target}}$ at $\{(p_i + v_i, v_i + u_i)\}_{i \in \{1,\ldots,n\}}$, is disconnected.

Proof. The proof of the statement for $\mathcal{G}_{\mathrm{target}} = \mathcal{G}_{\mathrm{disk}}(r_{\mathrm{cmm}})$ is straightforward. Consider two agents whose relative position and velocity belong to $\mathcal{A}_0^d \setminus \mathcal{A}_1^d$. Then, after one time step, the two agents will not be connected in $\mathcal{G}_{\mathrm{disk}}(r_{\mathrm{cmm}})$ no matter what their controls are. Next, consider the case $\mathcal{G}_{\mathrm{target}} = \mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, 2r_{\mathrm{ctr}})$. For $d = 1$, let $\bar{v}$ be the maximal velocity in $\mathcal{A}^1(r_{\mathrm{cmm}}, 2r_{\mathrm{ctr}})$ at $p = 0$, that is, $\bar{v} = \min\{r_{\mathrm{cmm}}/m + (m-1)r_{\mathrm{ctr}} \mid m \in \mathbb{N}\}$. Take three agents with positions $p_1 = p_2 = p_3 = 0$ and velocities $v_1 = -\bar{v}$, $v_2 = 0$, and $v_3 = \bar{v}$. At this configuration, the graph $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, 2r_{\mathrm{ctr}})$ contains two edges: between agent 1 and 2 and between agent 2 and 3. Connectivity is preserved after one time step if agent 2 remains connected to both agents 1 and 3 after one time step. However, to remain connected with agent 1, its control u_2 must be equal to $-r_{\mathrm{ctr}}$ and, analogously, to remain connected with agent 3, its control u_2 must be equal to $+r_{\mathrm{ctr}}$. Clearly this is impossible. ∎

REMARKS 3.4.

(i) *The result in Lemma 3.3 on the double integrator graph has the following interpretation. Assume that agent i has two neighbors j and k in the graph $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, r_{\mathrm{ctr}})$. By definition of the neighboring law for this graph, we know that there exists bounded controls for i and j to maintain the $((p_i, v_i), (p_j, v_j))$ link and that there exists bounded controls for i and k to maintain the $((p_i, v_i), (p_k, v_k))$ link. The lemma states that, for some states of the agents i, j, and k, there might not exist controls that maintain both links simultaneously.*

(ii) *In other words, Lemma 3.3 shows how the disk graph $\mathcal{G}_{\mathrm{disk}}(r_{\mathrm{cmm}})$ and the double integrator disk graph $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, 2r_{\mathrm{ctr}})$ are not appropriate choices for the connectivity maintenance problem.* □

The following theorem suggests that an appropriate scaling of the control bound is helpful in identifying a suitable state dependent graph for Problem 3.2.

THEOREM 3.5 (A scaled double-integrator disk graph is suitable). *Consider a network of n agents with double integrator dynamics (3.1) in $\mathbb{R}^d$. Let r_{cmm} be the communication range and let r_{ctr} be the control bound. For $k \in \{1, \ldots, n-1\}$, define*

$$\nu(k) = \frac{2}{k\sqrt{d}}.$$

Assume that $k \in \{1, \ldots, n-1\}$ and the state $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$ together have the property that the graph $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}})$ at $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$ contains a spanning tree T with diameter at most k. Then there exists $\{u_i\}_{i \in \{1,\ldots,n\}} \subseteq B(0_d, r_{\mathrm{ctr}})$ such

that if $((p_i, v_i), (p_j, v_j))$ is an edge of T, then $((p_i + v_i, v_i + u_i), (p_j + v_j, v_j + u_j))$ is an edge of $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$ at $\{(p_i + v_i, v_i + u_i)\}_{i \in \{1,\ldots,n\}}$.

These results are based upon Shostak's Theory for systems of inequalities, as exposed in [9]. We provide the proof in Appendix B. The following results are immediate consequences of this theorem.

COROLLARY 3.6 (Simple sufficient condition). *With the same notation in Theorem 3.5, define* $\nu_{\min} = \frac{2}{(n-1)\sqrt{d}}$. *Assume that the state* $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$ *has the property that the graph* $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu_{\min} r_{\text{ctr}})$ *is connected at* $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$. *Then*

    (i) *there exists* $\{u_i\}_{i \in \{1,\ldots,n\}} \subseteq B(0_d, r_{\text{ctr}})$, *such that the graph* $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu_{\min} r_{\text{ctr}})$ *is also connected at* $\{(p_i + v_i, v_i + u_i)\}_{i \in \{1,\ldots,n\}}$; *and*

    (ii) *if* T *is a spanning tree of* $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu_{\min} r_{\text{ctr}})$ *at* $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$, *then there exists* $\{u_i\}_{i \in \{1,\ldots,n\}} \subseteq B(0_d, r_{\text{ctr}})$, *such that, for all edges* $((p_i, v_i), (p_j, v_j))$ *of* T, *it holds that* $((p_i + v_i, v_i + u_i), (p_j + v_j, v_j + u_j))$ *is an edge of* $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu_{\min} r_{\text{ctr}})$ *at* $\{(p_i + v_i, v_i + u_i)\}_{i \in \{1,\ldots,n\}}$.

REMARK 3.7 (Scaling of $\nu_{\min}$ with n). *The condition* $\nu_{\min} = \frac{2}{\sqrt{d}(n-1)}$ *implies that according to the sufficient conditions in Corollary 3.6, as the number of agents grows, the velocities of the agents must be closer and closer in order for the agents to be able to remain connected.*

If $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$ *at* $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$ *is not connected for some* k, *then Theorem 3.5 applies to its connected components. In what follows we fix* k *and without loss of generality assume the graph* $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$ *at* $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$ *to be connected.* □

REMARK 3.8 (Distributed computation of connectivity and of spanning trees). *Each agent can compute its neighbors in the graph* $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$ *just by communicating with its neighbors in* $\mathcal{G}_{\text{disk}}(r_{\text{cmm}})$ *and exchanging with them position and velocity information. Alternatively, this computation may also be performed if each agent may sense relative position and velocity with all other agents at a distance no larger than* r_{cmm}.

It is possible to compute spanning trees in networks via standard depth-first search distributed algorithms. It is also possible [10] to distributively compute a minimum diameter spanning tree in a network. □

3.4. The control constraint set and its polytopic representation. We now concentrate on two agents with indices i and j. For $t \in \mathbb{N}_0$, we define the relative position, velocity and control by $p_{ij}[t] = p_i[t] - p_j[t]$, $v_{ij}[t] = v_i[t] - v_j[t]$ and $u_{ij}[t] = u_i[t] - u_j[t]$, respectively. It is easy to see that

$$p_{ij}[t + 1] = p_{ij}[t] + v_{ij}[t],$$
$$v_{ij}[t + 1] = v_{ij}[t] + u_{ij}[t].$$

Assume that agents i, j are connected in $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$ at time t. By definition, this means that the relative state $(p_{ij}[t], v_{ij}[t])$ belongs to $\mathcal{A}^d(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$. If this connection is to be maintained at time $t + 1$, then the relative control at time t must satisfy

$$u_i[t] - u_j[t] \in \mathcal{U}^d(r_{\text{cmm}}, \nu(k)r_{\text{ctr}}) \cdot (p_{ij}[t], v_{ij}[t]). \tag{3.2}$$

Also, implicit are the following bounds on individual control inputs $u_i[t]$ and $u_j[t]$:

$$u_i[t] \in B(0_d, r_{\text{ctr}}), \quad u_j[t] \in B(0_d, r_{\text{ctr}}). \tag{3.3}$$

This discussion motivates the following definition.

DEFINITION 3.9. *Given $r_{\mathrm{cmm}}, r_{\mathrm{ctr}}, \nu(k) \in \mathbb{R}_+$ and given a set E of edges in $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}})$ at $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$, the* control constraint set *is defined by*

$$
\begin{aligned}
\mathcal{U}_E^d(r_{\mathrm{cmm}}, r_{\mathrm{ctr}}, \nu(k)) \cdot (\{p_i, v_i\}_{i \in \{1,\ldots,n\}}) \\
= \{(u_1, \ldots, u_n) \in B(0_d, r_{\mathrm{ctr}})^n \mid \forall((p_i, v_i), (p_j, v_j)) \in E, \\
u_i - u_j \in \mathcal{U}^d(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}}) \cdot (p_i - p_j, v_i - v_j)\}.
\end{aligned}
$$

In other words, the control constraint set for an edge set E is the set of controls for each agent with the property that all edges in E will be maintained in one time step.

REMARK 3.10. *We can now interpret the results in Theorem 3.5 as follows.*
   (i) *To maintain connectivity between any pair of connected agents, we should simultaneously handle constraints corresponding to* all *edges of $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}})$. This might render the control constraint set empty.*
   (ii) *However, if we only consider constraints corresponding to edges belonging to a spanning tree T of $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}})$, then the set $\mathcal{U}_T^d(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}}) \cdot (\{p_i, v_i\}_{i \in \{1,\ldots,n\}})$ is guaranteed to be nonempty.* □

Let us now provide a concrete representation of the control constraint set. Given a pair i, j of connected agents, the admissible control set $\mathcal{U}^d(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}}) \cdot (p_{ij}, v_{ij})$ is convex and compact (Lemma 2.2). Hence, we can fit a polytope with N_{poly} sides inside it. This approximating polytope leads to the following tighter version of the constraint in (3.2):

$$
(C_{ij}^\eta)^T (u_i - u_j) \le w_{ij}^\eta, \qquad \eta \in \{1, \ldots, N_{\mathrm{poly}}\}, \tag{3.4}
$$

for some appropriate vector $C_{ij}^\eta \in \mathbb{R}^d$ and scalar $w_{ij}^\eta \in \mathbb{R}$. Similarly, one can compute an inner polytopic approximation of the closed ball $B(0_d, r_{\mathrm{ctr}})$ and write the following linear vector inequalities:

$$
(C_{i\theta}^\eta)^T u_i \le w_{i\theta}^\eta, \qquad \eta \in \{1, \ldots, N_{\mathrm{poly}}\}, \tag{3.5}
$$

where the symbol θ has the interpretation of a fictional agent. In this way, we have cast the original problem of finding a set of feasible control inputs into a satisfiability problem for a set of linear inequalities.

REMARK 3.11. *Rather than a network-wide control constraint set, one might like to obtain decoupled constraint sets for each individual agent. However, (1) it is not clear how to design a distributed algorithm to perform this computation, (2) such an algorithm will likely have large communication requirements, and (3) such a calculation might lead to a very conservative estimate for the decoupled control constraint sets. Therefore, rather than explicitly decoupling the control constraint sets, we next focus on a distributed algorithm to search the control constraint set for feasible controls that are optimal according to some criterion.* □

4. Distributed computation of optimal controls. In this section we formulate and solve the following optimization problem: given an array of desired control inputs $U_{\mathrm{des}} = (u_{\mathrm{des},1}, \ldots, u_{\mathrm{des},n})^T \in (\mathbb{R}^d)^n$, find, via local computation, the array $U = (u_1, \ldots, u_n)$ belonging to the control constraint set, that is *closest* to the desired array U_{des}. To formulate this problem precisely, we need some additional notations. Let E be a set of edges in the undirected graph $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}})$ at

$\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$. To deal with the linear inequalities of the form (3.4) and (3.5) associated to each edge of E, we introduce an appropriate *multigraph*. A *multigraph* (or *multiple edge graph*) is, roughly speaking, a graph with multiple edges between the same vertices. More formally, a multigraph is a pair $(V_{\text{mult}}, E_{\text{mult}})$, where V_{mult} is the vertex set and the edge set E_{mult} contains numbered edges of the form (i, j, η), for $i, j \in V$ and $\eta \in \mathbb{N}$, and where E_{mult} has the property that if $(i, j, \eta) \in E_{\text{mult}}$ and $\eta > 1$, then also $(i, j, \eta - 1) \in E_{\text{mult}}$. In what follows, we let G_{mult} denote the multigraph with vertex set $\{1, \ldots, n\}$ and with edge set $E_{\text{mult}} = \{(i, j, \eta) \in \{1, \ldots, n\}^2 \times \{1, \ldots, N_{\text{poly}}\} \mid ((p_i, v_i), (p_j, v_j)) \in E, \ i > j\}$. Note that to each element $(i, j, \eta) \in E_{\text{mult}}$ is associated the inequality $(C_{ij}^\eta)^T (u_i - u_j) \leq w_{ij}^\eta$. We are now ready to formally state the optimization problem at hand in the form of the following quadratic programming problem:

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2} \sum_{i=1}^{n} \| u_i - u_{\text{des},i} \|^2, \\
\text{subj. to} \quad & (C_{ij}^\eta)^T (u_i - u_j) \leq w_{ij}^\eta, \quad \text{for } (i, j, \eta) \in E_{\text{mult}}, \\
& (C_{i\theta}^\eta)^T u_i \leq w_{i\theta}^\eta, \qquad \text{for } i \in \{1, \ldots, n\}, \eta \in \{1, \ldots, N_{\text{poly}}\}.
\end{aligned}
\tag{4.1}
$$

Here, somehow arbitrarily, we have adopted the 2-norm to define the cost function.

REMARK 4.1 (Feasibility). *If E is a spanning tree of $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu r_{\text{ctr}})$ at a connected configuration $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$, then the control constraint set $\mathcal{U}_E^d(r_{\text{cmm}}, r_{\text{ctr}}, \nu(k)) \cdot (\{p_i, v_i\}_{i \in \{1,\ldots,n\}})$ is guaranteed to be non-empty by Theorem 3.5. In turn, this implies that the optimization problem (4.1) is feasible.* □

4.1. Solution via duality: the projected Jacobi method. The problem (4.1) can be written in a compact form as:

$$
\begin{aligned}
\text{minimize} \quad & \frac{1}{2} \| U - U_{\text{des}} \|^2, \\
\text{subj. to} \quad & B_{\text{mult}}^T U \preceq w,
\end{aligned}
$$

for appropriately defined matrix B_{mult} and vector w. A dual "projected Jacobi method" algorithm for the solution of this standard quadratic program is described in Appendix A. According to this algorithm, let λ^* be the value of Lagrange multipliers at optimality. Then the global minimum for U is achieved at

$$
U^* = U_{\text{des}} - B_{\text{mult}} \lambda^*.
\tag{4.2}
$$

The projected Jacobi iteration for each component of λ is given by

$$
\lambda_\alpha(t+1) = \max \Big\{ 0, \lambda_\alpha(t) - \frac{\tau}{(B_{\text{mult}}^T B_{\text{mult}})_{\alpha\alpha}} \Big((w - B_{\text{mult}}^T U_{\text{des}})_\alpha
$$
$$
+ \sum_{\beta=1}^{N_{\text{poly}}(e+n)} (B_{\text{mult}}^T B_{\text{mult}})_{\alpha\beta} \lambda_\beta(t) \Big) \Big\}, \quad (4.3)
$$

where $\alpha \in \{1, \ldots, N_{\text{poly}}(e + n)\}$ and τ is the step size parameter. We can select $\tau = \frac{1}{N_{\text{poly}}(e+n)}$ to guarantee convergence.

4.2. A distributed implementation of the dual algorithm. Because of the particular structure of the matrix $B_{\text{mult}}^T B_{\text{mult}}$, the iterations (4.3) can be implemented in a distributed way over the original graph G. To highlight the distributed structure of the iteration we denote the components of λ by referring to the nodes that they share and the inequality they are related to. In particular for each edge in G_{mult}, the corresponding Lagrange multiplier will be denoted as λ_{ij}^η if the edge goes from node i to node j, $i > j$, and the edge is associated to the inequality constraint $C_{ij}^\eta(u_i - u_j) \leq w_{ij}^\eta$. This makes up the first $N_{\text{poly}}e$ entries of the vector λ. To be consistent with this notation, the next $N_{\text{poly}}n$ entries will be denoted $\lambda_{1\theta}^1, \ldots, \lambda_{1\theta}^{N_{\text{poly}}}, \ldots, \lambda_{n\theta}^1, \ldots, \lambda_{n\theta}^{N_{\text{poly}}}$. Additionally, define $\mathcal{N}(i) = \{j \in \{1, \ldots, n\} \mid \{(p_i, v_i), (p_j, v_j)\} \in E\} \cup \{\theta\}$. The symbol θ has the interpretation of a fictional node.

Defining $\lambda_{ij}^\eta := \lambda_{ji}^\eta$ and $C_{ij}^\eta := -C_{ji}^\eta$ for $i < j$, we can write equations (4.2) and (4.3) in components as follows. Equation (4.2) reads, for $i \in \{1, \ldots, n\}$,

$$u_i^* = u_{\text{des},i} - \sum_{k \in \mathcal{N}(i)} \sum_{\eta=1}^{N_{\text{poly}}} C_{ik}^\eta \lambda_{ik}^\eta.$$

One can easily work an explicit expression for matrix product $B_{\text{mult}}^T B_{\text{mult}}$ in (4.3). Then, equation (4.3) reads, for $(i, j, \eta) \in E_{\text{mult}}$,

$$\lambda_{ij}^\eta(t+1) = \max\left\{ 0, \lambda_{ij}^\eta(t) - \frac{\tau}{2(C_{ij}^\eta)^T C_{ij}^\eta} \cdot \right.$$
$$\left(\sum_{k \in \mathcal{N}(i)} \sum_{\sigma=1}^{N_{\text{poly}}} \left((C_{ij}^\eta)^T C_{ik}^\sigma \lambda_{ik}^\sigma \right) + \sum_{k \in \mathcal{N}(j)} \sum_{\sigma=1}^{N_{\text{poly}}} \left((C_{ji}^\eta)^T C_{jk}^\sigma \lambda_{jk}^\sigma \right) \right.$$
$$\left. \left. + w_{ij}^\eta - (C_{ij}^\eta)^T(u_{\text{des},i} - u_{\text{des},j}) \right) \right\},$$

together with, for $i \in \{1, \ldots, n\}$, $\eta \in \{1, \ldots, N_{\text{poly}}\}$,

$$\lambda_{i\theta}^\eta(t+1) = \max\left\{ 0, \lambda_{i\theta}^\eta(t) \right.$$
$$\left. - \frac{\tau}{(C_{i\theta}^\eta)^T C_{i\theta}^\eta} \left(\sum_{k \in \mathcal{N}(i)} \sum_{\sigma=1}^{N_{\text{poly}}} ((C_{i\theta}^\eta)^T C_{ik}^\sigma \lambda_{ik}^\sigma) + w_{i\theta}^\eta - (C_{i\theta}^\eta)^T u_{\text{des},i} \right) \right\}.$$

We distribute the task of running iterations for these $N_{\text{poly}}(e + n)$ Lagrange multipliers among the n agents as follows: an agent i carries out the updates for all quantities $\lambda_{i\theta}^\eta$ and all λ_{ij}^η for which $i > j$. By means of this partition and by means of iterated one-hop communication among agents, it is possible to compute the global solution for the optimization problem (4.1) in a distributed fashion over the double integrator disk graph.

5. Simulations. To illustrate our analysis we focus on the following scenario. For the two dimensional setting, i.e., for $d = 2$, we assume that there are $n = 5$ agents with (randomly chosen) initial condition and such that they are connected according to $\mathcal{G}_{\text{di-disk}}$. The bound for the control input is $r_{\text{ctr}} = 2$ and the communication radius is $r_{\text{cmm}} = 10$.

We assigned to one of the agents a derivative feedback control $u_x[p,v] = (v_x - 2)$, $u_y[p,v] = (v_y - 5)$ as desired input. For the other agents the desired input is set to zero. We show the agent trajectories in Figure 5.1a, the velocities of the agents with respect to time in Figure 5.1b, and the distances between agents which are neighbors in the spanning tree in Figure 5.1c. Notice that the agents move with approximately identical velocity reaching a configuration in which all of them are at the limit distance $r_{\mathrm{cmm}} = 10$. The interesting aspect of this simulation is that the maintenance of connectivity leads to the accomplishment of apparently unrelated coordination tasks as velocity alignment and cohesiveness. This numerical result illustrate how our connectivity maintenance approach might indeed be a starting point for novel investigations into the problem of flocking with connectivity.
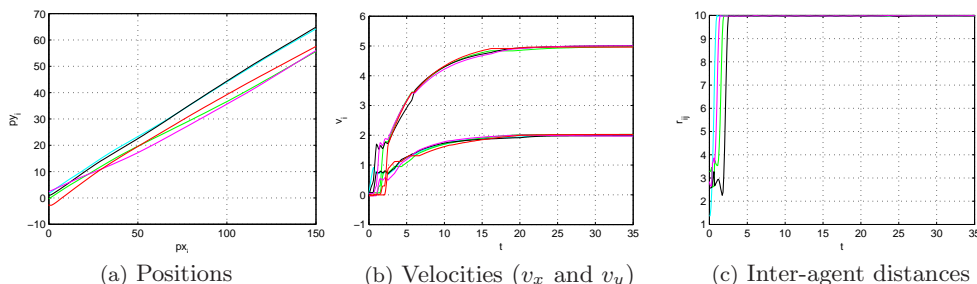

(a) Positions   (b) Velocities (v_x and v_y)   (c) Inter-agent distances

Fig. 5.1: Velocity alignment and cohesiveness for 5 agents in the plane ($d = 2$)

6. Conclusion. We provide some distributed algorithms to enforce connectivity among networks of agents with double-integrator dynamics. Future directions of research include (i) evaluating the communication complexity of the proposed distributed dual algorithm and possibly designing faster ones, (ii) studying the relationship between the connectivity maintenance problem and the platooning and mesh stability problem, and (iii) investigating the flocking phenomenon and designing flocking algorithms which do not rely on a blanket assumption of connectivity.

REFERENCES

[1] G. Notarstefano, K. Savla, F. Bullo, and A. Jadbabaie, "Maintaining limited-range connectivity among second-order agents," in *American Control Conference*, (Minneapolis, MN), pp. 2124–2129, June 2006.

[2] H. Ando, Y. Oasa, I. Suzuki, and M. Yamashita, "Distributed memoryless point convergence algorithm for mobile robots with limited visibility," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 5, pp. 818–828, 1999.

[3] J. Lin, A. S. Morse, and B. D. O. Anderson, "The multi-agent rendezvous problem," in *IEEE Conf. on Decision and Control*, (Maui, HI), pp. 1508–1513, Dec. 2003.

[4] J. Cortés, S. Martínez, and F. Bullo, "Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions," *IEEE Transactions on Automatic Control*, vol. 51, no. 8, pp. 1289–1298, 2006.

[5] A. Ganguli, J. Cortés, and F. Bullo, "On rendezvous for visually-guided agents in a nonconvex polygon," in *IEEE Conf. on Decision and Control and European Control Conference*, (Seville, Spain), pp. 5686–5691, Dec. 2005.

[6] D. P. Spanos and R. M. Murray, "Motion planning with wireless network constraints," in *American Control Conference*, (Portland, OR), pp. 87–92, June 2005.

[7] M. M. Zavlanos and G. J. Pappas, "Controlling connectivity of dynamic graphs," in *IEEE Conf. on Decision and Control and European Control Conference*, (Seville, Spain), pp. 6388–6393, Dec. 2005.

[8] Z. Gao, "On discrete time optimal control: A closed-form solution," in *American Control Conference*, (Boston, MA), pp. 52–58, 2004.

[9] B. Aspvall and Y. Shiloach, "A polynomial time algorithm for solving systems of linear inequalities with two variables per inequality," *SIAM Journal on Computing*, vol. 9, no. 4, pp. 827–845, 1980.

[10] M. Bui, F. Butelle, and C. Lavault, "A distributed algorithm for constructing a minimum diameter spanning tree," *Journal of Parallel and Distributed Computing*, vol. 64, pp. 571–577, 2004.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods.* Belmont, MA: Athena Scientific, 1997.

Appendix A. Projected Jacobi method. We briefly review here a parallel algorithm for the solution of a quadratic optimization problem. The technique is known as the *projected Jacobi method* in the literature on network flow control problems ([11], Section 3.4).

Consider the quadratic programming problem

$$\text{minimize} \quad \frac{1}{2}x^T Q x - b^T x,$$
$$\text{subj. to} \quad Ax \preceq c,$$

where Q is a given $n \times n$ symmetric positive definite matrix, A is a given $m \times n$ matrix, and $b \in \mathbb{R}^n$ and $c \in \mathbb{R}^m$ are given vectors. The dual problem is

$$\text{minimize} \quad \frac{1}{2}y^T F y + s^T y,$$
$$\text{subj. to} \quad y \succeq 0,$$

for $F = AQ^{-1}A^T$ and $s = c - AQ^{-1}b$. If y^* solves the dual problem, then $x^* = Q^{-1}(b - A^T y^*)$ solves the primal problem.

For a step size parameter $\tau > 0$ and for $j \in \{1, \ldots, n\}$, the projected Jacobi iteration, when the jth coordinate is updated, has the form

$$y_j(t+1) = \max\left\{0, y_j(t) - \frac{\tau}{f_{jj}}\left(s_j + \sum_{k=1}^{m} f_{jk}y_k(t)\right)\right\}, \qquad \text{(A.1)}$$

where f_{jk} is the j, k^{th} element of the matrix F. As discussed in [11], this algorithm converges to the global solution of the dual problem if the step size τ is chosen sufficiently small; in particular, convergence is guaranteed for $\tau = 1/m$.

Appendix B. Appendix on Shostak's test. This section provides a proof for Theorem 3.5. The proof amounts to showing that if E is the edge set of a spanning tree T in $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$ at $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$, then the control constraint set $\mathcal{U}_E^d(r_{\text{cmm}}, r_{\text{ctr}}, \nu(k)) \cdot (\{p_i, v_i\}_{i \in \{1,\ldots,n\}})$ is non-empty. We first consider a polytopic approximation of constraints (3.2) and (3.3). Among all possible choices, we use the conservative orthotope approximation that allows us to decouple the constraints into d independent sets of linear inequalities (one for each dimension). Then we use Shostak's theory to obtain sufficient conditions for the feasibility of these linear inequalities. For brevity, we drop the dependence of the quantities on t and we assume

that the variables u_i are scalars, for all $i \in \{1, \ldots, n\}$ and $t \geq 0$. The resulting sets of linear inequalities for one particular dimension are

$$\delta^l_{i,j} \leq u_i - u_j \leq \delta^u_{i,j}, \quad \text{and} \quad -\frac{r_{\mathrm{ctr}}}{\sqrt{d}} \leq u_i \leq \frac{r_{\mathrm{ctr}}}{\sqrt{d}}. \tag{B.1}$$

where $-\nu(k)r_{\mathrm{ctr}} \leq \delta^l_{i,j} \leq \delta^u_{i,j} \leq \nu(k)r_{\mathrm{ctr}}$, for all $i, j \in \{1, \ldots, n\}$ and $i \neq j$.

B.1. Shostak Theory. In this section we present Shostak's theory for feasibility of linear inequalities involving at most two variables, similar to the ones in (B.1). These ideas will then be used to prove Theorem 3.5. The notations used in [9] adapted to our case are presented next. Let u_0 be an auxiliary *zero variable* that always occurs with zero coefficient - the only variable that can do this. Without loss of generality, we can thus assume that all the inequalities in $\mathcal{L}$ contain two variables. As a result of this, the inequalities in (B.1) can be succinctly written as

$$u_i - u_j \leq \delta_{i,j}, \quad \forall i, j \in \{0, \ldots, n\}, \tag{B.2}$$

where for all $i, j \in \{1, \ldots, n\}$, $i \neq j, -\nu(k)r_{\mathrm{ctr}} \leq \delta_{i,j} \leq \nu(k)r_{\mathrm{ctr}}$ and for all $i \in \{1, \ldots, n\}, \delta_{i,0} = \delta_{0,i} = \frac{r_{\mathrm{ctr}}}{\sqrt{d}}$. Also implicit in this formulation is the relation that $\delta_{i,j} + \delta_{j,i} \geq 0$ for all $i, j \in \{0, \ldots, n\}$ and $i \neq j$.

Let $\mathcal{L}$ denote the system of inequalities in (B.2). We construct the graph $G(\mathcal{L})$ with $n + 1$ vertices and $2(2n - 1)$ edges as follows: (a) For each variable u_i occurring in $\mathcal{L}$, add a vertex i to $G(\mathcal{L})$. (b) For each inequality of the form $a_{i,j}u_i + b_{i,j}u_j \leq \delta_{i,j}$ in $\mathcal{L}$, add an undirected edge between i and j to $G(\mathcal{L})$, and label the edge with the inequality (see Figure B.1). It is easy to see the following relation between the spanning tree T in $\mathcal{G}_{\mathrm{di\text{-}disk}}(r_{\mathrm{cmm}}, \nu(k)r_{\mathrm{ctr}})$ at $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$ that is used to derive the constraints in the inequalities (B.2) and the graph $G(\mathcal{L})$: (a) The vertex set of $G(\mathcal{L})$ is the union of the vertex set of T and the auxiliary vertex 0 (b) For every edge $\{i, j\}$ in T, there are two edges between the vertices i and j in $G(\mathcal{L})$ (c) Additionally, $G(\mathcal{L})$ contains two edges between 0 and every other vertex i, for all $i \in \{1, \ldots, n\}$.
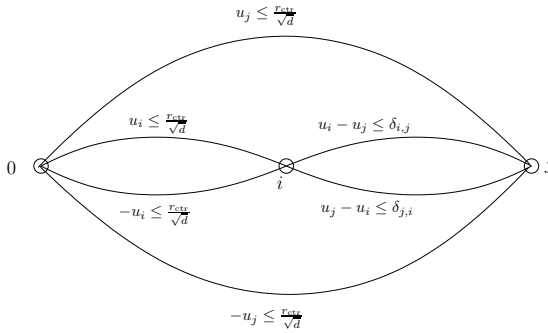

Fig. B.1: Snippet of the graph $G(\mathcal{L})$ for the system of inequalities in (B.2)

To every edge represented by the inequality of the form $a_{i,j}u_i + b_{i,j}u_j \leq \delta_{i,j}$, we associate *a triple* $\langle a_{i,j}, b_{i,j}, \delta_{i,j} \rangle$. Note that $\langle b_{i,j}, a_{i,j}, \delta_{i,j} \rangle$ is also a triple associated with the same edge. Without loss of generality, consider a path of $G(\mathcal{L})$ determined by the vertices $\{1, 2, \ldots, l + 1\}$ and the edges $e_{1,2}, e_{2,3}, \ldots, e_{l,l+1}$ between them. *A triple sequence*, P, associated with the path is defined as

$$\langle a_{1,2}, b_{1,2}, \delta_{1,2} \rangle, \langle a_{2,3}, b_{2,3}, \delta_{2,3} \rangle, \ldots, \langle a_{l,l+1}, b_{l,l+1}, \delta_{l,l+1} \rangle,$$

where, for $1 \le i \le l$, $a_{i,i+1}u_i + b_{i,i+1}u_j \le \delta_{i,i+1}$ is the inequality associated with the edge $e_{i,i+1}$. If $a_{i+1,i+2}$ and $b_{i,i+1}$ have opposite signs for $1 \le i < l$, then P is called *admissible*.

Define $\langle a_P, b_P, \delta_P \rangle$, the *residue* of P, as

$$\langle a_P, b_P, \delta_P \rangle = \langle a_{1,2}, b_{1,2}, \delta_{1,2} \rangle \odot \langle a_{2,3}, b_{2,3}, \delta_{2,3} \rangle \odot \ldots \odot \langle a_{l,l+1}, b_{l,l+1}, \delta_{l,l+1} \rangle,$$

where $\odot$ is the associativity binary operator defined on triples by

$$\langle a, b, \delta \rangle \odot \langle a', b', \delta' \rangle = \langle \kappa aa', -\kappa bb', \kappa(\delta a' - \delta'b) \rangle,$$
$$\text{where} \quad \kappa = a'/|a'|.$$

Intuitively, the operator $\odot$ takes two inequalities and derives a new inequality by eliminating a common variable; e.g., $ax+by \le \delta$ and $a'y+b'z \le \delta'$ imply $-aa'x+bb'z \le -(\delta a' - \delta'b)$ if $a < 0$ and $b > 0$. Note that the signs of a_P and $a_{1,2}$ agree, as do the signs of b_P and $b_{1,2}$.

A path is called a *loop* if the initial and final vertices are identical. (A loop is not uniquely specified unless its initial vertex is given.) If all the intermediate vertices of a path are distinct, the path is *simple*. An admissible triple sequence P associated with a loop with initial vertex x is *infeasible* if its residue satisfies $a_P + b_P = 0$ and $\delta_P < 0$. A loop which contains an infeasible triple sequence is called an *infeasible loop*. Thus if $G(\mathcal{L})$ has an infeasible loop, the system of inequalities $\mathcal{L}$ is unsatisfiable. However, the converse is not true in general. Next, we show how to extend $\mathcal{L}$ to an equivalent system $\mathcal{L}'$ such that $G(\mathcal{L}')$ has an infeasible simple loop if and only if $\mathcal{L}$ is unsatisfiable.

For each vertex i of $G(\mathcal{L})$ and for each admissible triple sequence P with $a_P+b_P \ne 0$ associated with a simple loop of $G(\mathcal{L})$ and initial vertex i, add a new inequality $(a_P + b_P)u_i \le \delta_P$ to $\mathcal{L}$. This new system $\mathcal{L}'$ is referred to as the *Shostak extension* of $\mathcal{L}$. We now state the necessary and sufficient condition on the extended system of inequalities $\mathcal{L}'$ for the satisfiability of the original system $\mathcal{L}$.

THEOREM B.1 (Shostak's Theorem [9]). *Let $\mathcal{L}'$ be the Shostak extension of $\mathcal{L}$. The system of inequalities $\mathcal{L}$ is satisfiable if and only if $G(\mathcal{L}')$ contains no infeasible simple loop.*

B.2. Satisfiability test. In this section we use the Shostak criterion to derive conditions for the satisfiability of the inequalities in (B.2).

LEMMA B.2. *Let $\mathcal{L}$ be the system of inequalities of the form (B.2) obtained by considering pairwise neighbors in a spanning tree T in $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$ at $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$. Then the Shostak extension of $\mathcal{L}$ is itself.*

Proof. Consider a simple loop of $G(\mathcal{L})$ with the initial vertex $i \in \{0, 1, \ldots, n\}$. Consider an admissible triple sequence P associated with the loop. Since $a_{i,j}, b_{i,j} \in \{-1, +1\}$, for all $i, j \in \{1, \ldots, n\}, i \ne j$, and $a_{0,i}, a_{i,0}, b_{i,0}, b_{0,i} \in \{-1, 0, +1\}$, for all $i \in \{1, \ldots, n\}$, the residue of P, $\langle a_P, b_P, \delta_P \rangle$, is such that $a_p + b_p = 0$. Hence, no new inequality must be added to obtain the Shostak extension of $\mathcal{L}$. $\square$

LEMMA B.3. *Let $\mathcal{L}$ be the system of inequalities of the form (B.2) obtained by considering pairwise neighbors in a spanning tree T of depth at most k in $\mathcal{G}_{\text{di-disk}}(r_{\text{cmm}}, \nu(k)r_{\text{ctr}})$ at $\{(p_i, v_i)\}_{i \in \{1,\ldots,n\}}$. If $\nu(k) = \frac{2}{k\sqrt{d}}$, then there is no infeasible simple loop in $G(\mathcal{L})$.*

Proof. Looking at figure B.1 it is clear that there are two types of simple loops with admissible triple sequences in $G(\mathcal{L})$:

(i) $\langle +1, -1, \delta_{i,j} \rangle$, $\langle +1, -1, \delta_{j,i} \rangle$ or $\langle -1, +1, \delta_{i,j} \rangle$, $\langle -1, +1, \delta_{j,i} \rangle$,
where $i, j \in \{0, \ldots, n-1\}$ and $\{i, j\}$ is an edge in T.

(ii) $\langle 0, -1, \frac{r_{\text{ctr}}}{\sqrt{d}} \rangle, \langle +1, -1, \delta_{i_1, i_2} \rangle, \ldots, \langle +1, -1, \delta_{i_{l-1}, i_l} \rangle, \langle +1, 0, \frac{r_{\text{ctr}}}{\sqrt{d}} \rangle$ or
$\langle 0, +1, \frac{r_{\text{ctr}}}{\sqrt{d}} \rangle, \langle -1, +1, \delta_{i_2, i_1} \rangle, \ldots, \langle -1, +1, \delta_{i_l, i_{l,l-1}} \rangle, \langle -1, 0, \frac{r_{\text{ctr}}}{\sqrt{d}} \rangle$,
where $i_l \in \{1, \ldots, \zeta\}$ for all $l \in \{1, \ldots, \zeta\}$ and $\{i_l, i_{l+1}\}$ is an edge in T.

The residue for the first set of loops is $\langle +1, -1, \delta_{i,j} + \delta_{j,i} \rangle$ or $\langle -1, +1, \delta_{i,j} + \delta_{j,i} \rangle$. The feasibility condition is trivially satisfied by construction since $\delta_{i,j} + \delta_{j,i} \geq 0$. For the second set of loops, the residue is:

$$\left\langle 0, -1, \frac{r_{\text{ctr}}}{\sqrt{d}} \right\rangle \odot \langle +1, -1, \delta_{i_1, i_2} \rangle \odot \ldots \odot \langle +1, -1, \delta_{i_{\zeta-1}, i_\zeta} \rangle \odot \left\langle +1, 0, \frac{r_{\text{ctr}}}{\sqrt{d}} \right\rangle$$
$$= \left\langle 0, 0, 2\frac{r_{\text{ctr}}}{\sqrt{d}} + \sum_{l=1}^{\zeta-1} \delta_{i_l, i_{l+1}} \right\rangle,$$

or

$$\left\langle 0, +1, \frac{r_{\text{ctr}}}{\sqrt{d}} \right\rangle \odot \langle -1, +1, \delta_{i_2, i_1} \rangle \odot \ldots \odot \langle -1, +1, \delta_{i_\zeta, i_{\zeta-1}} \rangle \odot \left\langle -1, 0, \frac{r_{\text{ctr}}}{\sqrt{d}} \right\rangle$$
$$= \left\langle 0, 0, 2\frac{r_{\text{ctr}}}{\sqrt{d}} + \sum_{l=1}^{\zeta-1} \delta_{i_l, i_{l+1}} \right\rangle.$$

In order to guarantee the feasibility of the second set of loops, we need that $2\frac{r_{\text{ctr}}}{\sqrt{d}} + \sum_{l=1}^{\zeta-1} \delta_{i_l, i_{l+1}} \geq 0$. We derive conditions for the worst case which occurs when the loop is written for the longest path in T, i.e., when $\zeta = k+1$ and when $\delta_{i_l, i_{l+1}} = -\nu(k) r_{\text{ctr}}$, for all $l \in \{1, \ldots, k\}$. In this case, there is no infeasible simple loop if and only if

$$2\frac{r_{\text{ctr}}}{\sqrt{d}} - k\nu(k)r_{\text{ctr}} \geq 0,$$

that is, if and only if $\nu(k) = \frac{2}{k\sqrt{d}}$. □

Finally, the proof of Theorem 3.5 follows from Theorem B.1, Lemma B.2 and Lemma B.3.